TERMINAL DE TELECOMMUNICATION A DEUX ESPACES D'EXECUTION

L'invention concerne l'exécution de programmes et applications, sur un dispositif informatique à interface utilisateur (clavier, écran, carte son, zone tactile, souris, etc...), par exemple sur une passerelle domestique, une machine de vente (machine publique par exemple) ou encore sur un terminal de télécommunications (du PC – ordinateur personnel – au téléphone mobile).

On connaît différentes approches pour l'implémentation d'applications dans les terminaux de télécommunications.

Ainsi, le profil MIDP 2.0., sur machine virtuelle, met en œuvre une police de sécurité basée sur des standards ouverts, simple à utiliser, qui ne demande rien à l'utilisateur, qui prend en compte les besoins de chaque intervenant, du développement à l'exécution (on sépare les concepts d'utilisateur, d'opérateur, d'OEM, de tierce partie de confiance).

Il permet de protéger l'intégrité et vérifier l'origine des applications pendant le téléchargement et l'exécution de celles-ci, de contrôler l'accès aux ressources critiques suivant une politique de sécurité, d'avertir l'utilisateur de ce qui se passe, et peut même lui demander son avis.

La prise en compte de la politique de sécurité se fait assez simplement, au niveau d'une API à protéger, par un appel à la méthode « check permission » de la classe « Midlet » (figure 2).

Ceci demande que la fonction d'appel de fichier MIDP ne soit pas accessible directement depuis les programmes MIDP (fonction protected).

La politique de sécurité de MIDP 2.0 est très adaptée aux besoins des divers intervenants. La possibilité de demander l'avis de l'utilisateur suivant un certain nombre de critères (toujours, une fois, pour une session, jamais) est très avantageuse.

Il n'en reste pas moins que son implémentation pose deux types de problèmes.

Tout d'abord, l'exécution de la procédure protégée se fait dans le même espace d'exécution que le programme appelant, ce qui facilite le risque de « fuites ». Imaginons ainsi un service de chiffrement appelé par

deux midlets simultanément, rien ne garantit si on n'y fait guère attention qu'une midlet ne pourra récupérer le contenu de la clé privée utilisée par l'autre midlet.

Le premier problème est donc un manque de sécurité, notamment pour les applications à risque telles que le paiement, la signature ou encore par exemple les applications DRM.

Des exploits ont d'ailleurs montré qu'avec une erreur d'implémentation, on pouvait passer outre ce système de permissions.

Le deuxième problème du profil MIDP est posé par les spécifications du profil MIDP lui-même. Il n'est pas adapté à la preuve formelle de programmes. Cela pose un problème dans certains secteurs (notamment banquiers) où une midlet ne peut être modélisée par des méthodes formelles, et donc ne peut pas être certifiée par ces méthodes.

En d'autres termes, il n'existe pas de technique permettant de prouver, par des méthodes formelles, la validité par rapport à des spécifications d'un programme programmé dans ce profil.

Un autre profil, le profil STIP, est lui plus spécialement adapté pour donner accès à des APIs orientés quant à la sécurité, telles que l'accès à la SIM.

Les machines virtuelles STIP (figure 3) permettent de faire fonctionner les programmes spécialement écrits pour le profil STIP.

L'autre force de STIP est que son modèle de programmation et ses APIs se prêtent bien à l'analyse suivant les méthodes formelles. Cela a permis d'ailleurs de rallier le milieu bancaire à son design, car la conformité du code aux spécifications peut être prouvée de manière formelle.

Aussi le profil STIP, utilisé dans le milieu bancaire, est, par ses limitations, adapté à la preuve de programmes.

Toutefois, le profil STIP a été conçu pour des systèmes fermés (on n'y télécharge pas impunément des applications dont on n'a pas confiance).

Ainsi aucun modèle de sécurité n'est mis en place (dans la version 2.1.1. de la spécification), et donc toute application STIP (stiplet) peut accéder à n'importe quelle API de type STIP déjà implémentée.

Le profil STIP n'est donc pas adapté pour la réalisation de terminaux où un utilisateur est susceptible de télécharger et mettre en œuvre des applications courantes telles que jeux ou applications utilitaires divers.

L'invention se donne ici pour but de proposer une configuration permettant, dans un terminal de télécommunications, de mettre en œuvre à la fois des applications d'utilisateur diverses, et à la fois des applications requerrant un haut niveau de sécurité.

L'invention vise en outre à faciliter la programmation et la mise en œuvre d'applications, notamment en facilitant la certification du bon fonctionnement des applications nouvellement programmées.

On connaît certes le principe des téléphones mobiles hébergeant deux machines virtuelles sous la forme physique de deux processeurs, l'un constitués par le terminal lui-même et l'autre constitué par la carte SIM.

La carte SIM vérifie des exigences de sécurité élevées, tandis que le processeur du terminal lui-même et son contenu sont, eux, accessibles par l'utilisateur.

Toutefois une telle implémentation présente encore certains inconvénient majeurs.

Ainsi un autre but de l'invention est de proposer un dispositif, associé en réseau ou pas, dans lequel un espace sécurisé et un espace non sécurisé sont tous deux mis à profit, par exemple en permettant à l'espace sécurisé d'accéder aux interfaces utilisateurs telles que clavier ou écran à la place de l'espace non sécurisé, et a contrario permettre par exemple à l'espace non sécurisé d'accéder à une communication sécurisée avec un opérateur connu pour garantir une telle sécurité. On citera notamment, en tant que tel opérateur de sécurité, les opérateurs de téléphonie, notamment mobile, les banques, les fournisseurs d'objets multimédia à diffusion sélective ou payante, les opérateurs de fourniture de service contre signature électronique via ledit dispositif.

Des fournisseurs d'objets multimédia à diffusion sélective sont notamment les « DRM » (Digital Rights Management, gestion des droits sous licence), serveurs qui délivrent un contenu typiquement musical,vidéo

ou de jeu, sous licence, et sous la forme d'un fichier prévu pour être lu sous diverses contraintes, par exemple un certain nombre de fois.

Un but de l'invention est de proposer de tels moyens dans lesquels, en outre, on puisse être certain que les deux espaces d'exécution associés (l'un à sécurité plus élevée que l'autre) soient effectivement ceux qui étaient destinés ou autorisé à être associés l'un à l'autre ab initio.

Ces buts sont atteints selon l'invention grâce à un dispositif informatique à interface utilisateur, comprenant des moyens de mise en œuvre d'une série d'applications, ces moyens incluant notamment un espace d'exécution machine virtuelle/profil de fonctionnement, le dispositif comportant un second espace d'exécution machine virtuelle/profil de fonctionnement se distinguant du premier par au moins sa machine virtuelle ou son profil de fonctionnement, chaque espace d'exécution hébergeant des applications, les applications du second espace d'exécution étant des applications à degré de sécurité spécifiquement plus élevé que celui des applications du premier espace d'exécution du fait que les applications du premier espace d'exécution sont des applications mises en places et activées par l'utilisateur du terminal tandis que les applications du second espace d'exécution sont des applications non modifiables par l'utilisateur du terminal, caractérisé en ce que les deux espaces d'exécution sont hébergés par un moyen physique de traitement qui est agencé pour être non scindable en deux parties sans destruction de ce moyen physique de traitement.

On propose également selon l'invention un procédé de mise en œuvre d'applications au sein d'un dispositif informatique à interface utilisateur, le procédé faisant appel à des moyens de mise en œuvre d'une série d'applications, ces moyens incluant notamment un espace d'exécution machine virtuelle/profil de fonctionnement et un second espace d'exécution machine virtuelle/profil de fonctionnement se distinguant du premier par au moins sa machine virtuelle ou son profil de fonctionnement, chaque espace d'exécution hébergeant des applications, les applications du second espace d'exécution étant des applications à degré de sécurité spécifiquement plus élevé que celui des applications du premier espace d'exécution du fait que

les applications du premier espace d'exécution sont des applications mises en place et activées par l'utilisateur du terminal tandis que les applications du second espace d'exécution sont des applications non modifiables par l'utilisateur du terminal, caractérisé en ce que les deux espaces d'exécution sont hébergés par un moyen physique de traitement qui est agencé pour être non scindable en deux parties sans destruction de ce moyen physique de traitement.

D'autres caractéristiques, buts et avantages de l'invention apparaîtront à la lecture de la description détaillée qui va suivre, faite en référence aux figures annexées sur lesquelles :

- la figure 1 est un schéma illustrant une implémentation MIDP selon l'art antérieur ;

- la figure 2 est un schéma illustrant la mise en œuvre de moyens de protection dans une telle implémentation MIDP ;

- la figure 3 est un schéma illustrant une implémentation STIP, conforme à l'art antérieur ;

- la figure 4 illustre une configuration fonctionnelle d'un terminal conforme à l'invention, selon une variante préférentielle.

Le mode particulier de réalisation que l'on décrira maintenant permet de bénéficier du meilleur des deux techniques MIDP et STIP, données à titre d'exemple, au sein d'un environnement d'exécution cohérent.

On y constitue le profil « utilisateur », le profil MIDP. Ce profil est très populaire dans le monde mobile pour la création de jeux et d'applications utilitaires diverses. L'utilisateur peut ainsi télécharger et exécuter des applications qu'il trouve sur le réseau, comme il le ferait avec un téléphone MIDP courant. Le profil MIDP inclut donc ici des applications mises en place et activées par l'utilisateur lui-même.

Le profil STIP constitue ici un profil additionnel, et plus spécifiquement un profil « opérateur ». Le profil STIP est très adapté aux applications qui demandent un haut niveau de sécurité, telles que les applications bancaires. Ainsi, les consortiums bancaires ont déjà fait confiance en la possibilité de certifier des applications STIP par des

méthodes formelles pour les implémenter dans des terminaux de paiement électronique (TPE).

La présente invention permet donc de fournir aux développeurs un ensemble d'API opérateur dont l'exécution est assurée dans un espace d'exécution approprié à la programmation aisée par ces développeurs, espace de même profil ou non, totalement distinct.

Ce mode de réalisation permet donc à l'opérateur de fournir un lot d'applications sécurisées, telles que paiement, de signature ou encore de DRM, totalement indépendant du profil d'exécution des applications « courantes ».

Le terminal représenté en figure 4 inclut et fait fonctionner en harmonie deux machines virtuelles 100 et 200 de profil distincts P1 et P2 (ou non). L'une 100 des deux machines est dédiée aux applications utilisateurs, l'autre 200 aux applications de l'opérateur.

Les profils correspondants P1 et P2, ici respectivement le profil MIDP et le profil STIP, sont ici eux-mêmes dédiés respectivement aux applications utilisateur et aux applications opérateur.

A la figure 4, on a représenté ainsi deux machines virtuelles 100 et 200.

La machine virtuelle « utilisateur » 100 est mise à disposition de l'utilisateur pour télécharger, installer, désinstaller, exécuter, stopper, comme bon lui semble des applications dans le profil MIDP. Les applications 110 qui y tournent utilisent l'API 120 de ce profil, ainsi qu'une API « stub » au même profil que celui de la machine 100, cette API stub étant référencée 130 sur la figure 4.

La deuxième machine, référencée 200, est la machine virtuelle « opérateur » : seul l'opérateur, par exemple l'opérateur de téléphonie mobile ou encore l'opérateur internet (fournisseur d'accès), par un mécanisme OTA (Over The Air), peut administrer cet espace d'exécution.

Il peut y installer, désinstaller, activer, désactiver comme bon lui semble des applications 210 écrites suivant le formalisme du profil 100. Ces applications 210 ont accès aux APIs 220 du profil P2 et à une ou plusieurs API de haut niveau illustrées sous la référence 230 sur la figure 4.

Ces API de haut niveau 230 permettent d'accéder à des services offerts par le profil de la machine 100. L'accès aux APIs, que ce soit du profil de la machine 200 ou du stub 230 au profil de la machine 100, se fait suivant le modèle de sécurité inhérent au profil de la machine 200.

L'API « stub » 130 est une API de haut niveau, exprimée suivant le modèle de programmation du profil 100, permettant d'accéder à des services offerts par le profil P2. L'accès aux APIs, que ce soit au profil de la machine 100 ou d'un stub 130 se fait suivant le modèle de sécurité inhérent au profil P1 de la machine 100.

Le fonctionnement des stubs 130 et 230 est le suivant :

L'appel à une API du stub 130, 230 est transformé en flux d'octets (processus de sérialisation, ou marshalling/unmarshalling suivant les appellations).

Ce flux est reçu par un manager 140, 240 du profil opposé via un canal de communication 300, déserialisé et converti en l'exécution d'une procédure dans le profil distant. Le retour d'exécution de cette procédure est de nouveau sérialisé dans le profil distant, et repasse dans le canal de communication 300 entre les deux profils P1 et P2 des machines 100 et 200, la réponse est désérialisée dans le profil originel et transformée en retour d'appel de l'API « stub ».

Ainsi, on dispose de deux espaces d'exécution indépendants consistant ici chacun en une machine différente et un profil différent, et en relation très étroite par l'intermédiaire d'API stub 130 et 230.

En variante, les deux profils P1 et P2 peuvent être du même type, par exemple deux profils MIDP ou deux profils STIP pour deux machines différentes.

On notera également que l'on peut adopter deux profils P1 et p2 différents au sein d'une même machine virtuelle.

Ce mode de réalisation permet donc d'offrir une API de paiement aux développeurs d'applications MIDP, où le paiement lui-même s'effectuera dans le cadre de l'exécution d'une machine virtuelle STIP contrôlée par l'opérateur.

En d'autres termes, une application MIDP, développé aisément, pourra offrir un moyen de paiement à l'utilisateur en faisant fonctionner une application de paiement de la machine 200 via le lien de communication 300. Une application MIDP est donc, grâce à l'invention, capable d'offrir une fonctionnalité de paiement de grande fiabilité.

Les deux espaces d'exécution, 100 et 200, chacun constitué par un couple machine virtuelle/profil d'exécution, différents l'un de l'autre par le profil ou par la machine virtuelle, sont toutefois tous deux implémentés par une même dispositif physique de traitement 400 (même entité hardware 400).

Ce dispositif de traitement hébergeant les deux espaces d'exécution est unique en ce sens qu'il ne peut être scindé sans destruction de son fonctionnement.

Ainsi il est impossible de séparer physiquement les deux espaces d'exécution, et donc également impossible d'associer un espace ainsi séparé avec un autre espace, lui non autorisé.

Un telle réalisation sur moyen unique est obtenue par exemple en implémentant les deux espaces d'exécution sur un même circuit intégré formant un unique processeur.

On assure ainsi que deux environnements, l'un sécurisé et l'autre non sécurisé, sont indissociables.

La sécurité proposée par un opérateur (téléphonie, banque, administration à signature, diffuseur multimedia) s'en trouve améliorée, qu'il s'agisse d'empêcher des détournements de fonctions de paiement, d'assurer la confidentialité ou la non falsification de codes secrets, de fiabiliser l'usage d'une signature électronique, ou encore de veiller à empêcher de dépasser des droits d'utilisation limités d'œuvres payantes.

Avantageusement, les profils P1, P2 de chacun des deux espaces d'exécution 100, P1, 200, P2 sont respectivement un profil STIP et un profil faisant partie du groupe constitué des profils STIP, MIDP, OSGI, et « .net ».

REVENDICATIONS

1. Dispositif informatique à interface utilisateur, comprenant des moyens de mise en œuvre d'une série d'applications, ces moyens incluant notamment un espace d'exécution machine virtuelle/profil de fonctionnement (100, P1, 200, P2), le dispositif comportant un second espace d'exécution machine virtuelle/profil de fonctionnement (100, P1, 200, P2) se distinguant du premier par au moins sa machine virtuelle (100, 200) ou son profil de fonctionnement (P1, P2), chaque espace d'exécution hébergeant des applications (110, 120, 130, 140, 220, 230), les applications du second espace d'exécution (100, P1, 200, P2) étant des applications à degré de sécurité spécifiquement plus élevé que celui des applications du premier espace d'exécution (100, P1, 200, P2) du fait que les applications (110, 120, 130, 210, 220, 230) du premier espace d'exécution (100, P1, 200, P2) sont des applications modifiables par l'utilisateur tandis que les applications (110, 120, 130, 210, 220, 230) du second espace d'exécution (100, P1, 200, P2) sont des applications non modifiables par l'utilisateur, caractérisé en ce que les deux espaces d'exécution sont hébergés par un même moyen physique de traitement (400) qui est agencé pour être non scindable en deux parties sans destruction de ce moyen physique de traitement (400).

2. Dispositif selon la revendication 1, caractérisé en ce que les applications (110, 120, 130, 210, 220, 230) du second espace d'exécution (100, P1, 200, P2) sont des applications modifiables par un opérateur de sécurité appartenant au groupe constitué des opérateurs de téléphonie, banques, fournisseurs d'objets multimédia à diffusion sélective ou payante, opérateurs de fourniture de services contre signature électronique via ledit dispositif.

3. Dispositif selon la revendication 1 ou la revendication 2, caractérisé en ce qu'il constitue un terminal téléphonique.

4. Dispositif selon la revendication 3, caractérisé en ce qu'il constitue un terminal de téléphonie mobile.

5. Dispositif selon l'une quelconque des revendications précédentes, caractérisé en ce qu'il comporte des moyens de communication (130, 230, 300) entre les deux espaces d'exécution (100, P1, 200, P2).

6. Dispositif selon la revendication 5, caractérisé en ce que les moyens de communication (130, 230, 300) entre les deux espaces d'exécution sont prévus pour autoriser une application (130, 230) d'un des deux espaces d'exécution à faire appel à des moyens de traitement du second espace d'exécution (100, P1, 200, P2).

7. Dispositif selon l'une quelconque des revendications précédentes, caractérisé en ce que chacun des deux espaces d'exécution inclut au moins une API distincte (120, 130, 220, 230).

8. Dispositif selon l'une quelconque des revendications précédentes, caractérisé en ce que les moyens de communication incluent une API « stub » (130, 230) ayant pour rôle de faire appel à des ressources de l'espace d'exécution opposé (100, P1, 200, P2), ces ressources mettant en œuvre une sélection quant à l'accès à elles-mêmes en fonction de l'application (110, 210) les appelant.

9. Dispositif selon l'une quelconque des revendications précédentes, caractérisé en ce que les moyens de communication entre les deux espaces d'exécution (100, P1, 200, P2) incluent des moyens mettant en œuvre une sérialisation/désérialisation ou un marshalling/unmarshalling.

10. Dispositif selon l'une quelconque des revendications précédentes, caractérisé en ce que l'un des deux espaces d'exécution (100, P1, 200, P2) inclut un profil de type STIP.

11. Dispositif selon l'une quelconque des revendications précédentes, caractérisé en ce que l'un des deux espaces d'exécution (100, P1, 200, P2) inclut un profil MIDP.

12. Dispositif selon l'une quelconque des revendications précédentes, caractérisé en ce que les profils (P1, P2) de chacun des deux espaces d'exécution (100, P1, 200, P2) sont respectivement un profil STIP

et un profil faisant partie du groupe constitué des profils STIP, MIDP, OSGI, et « .net ».

13. Procédé de mise en œuvre d'applications au sein d'un dispositif informatique à interface utilisateur, le procédé faisant appel à des moyens de mise en œuvre d'une série d'applications, ces moyens incluant notamment un espace d'exécution machine virtuelle/profil de fonctionnement (100, P1, 200, P2) et un second espace d'exécution machine virtuelle/profil de fonctionnement (100, P1, 200, P2) se distinguant du premier par au moins sa machine virtuelle (100, 200) ou son profil de fonctionnement (P1, P2), chaque espace d'exécution (100, P1, 200, P2) hébergeant des applications, les applications du second espace d'exécution (100, P1, 200, P2) étant des applications à degré de sécurité spécifiquement plus élevé que celui des applications du premier espace d'exécution (100, P1, 200, P2) du fait que les applications (110, 120, 130, 210, 220, 230) du premier espace d'exécution (100, P1, 200, P2) sont des applications modifiables par l'utilisateur tandis que les applications (110, 120, 130, 210, 220, 230) du second espace d'exécution (100, P1, 200, P2) sont des applications non modifiables par l'utilisateur, caractérisé en ce que les deux espaces d'exécution sont hébergés par un même moyen physique de traitement (400) qui est agencé pour être non scindable en deux parties sans destruction de ce moyen physique de traitement (400).

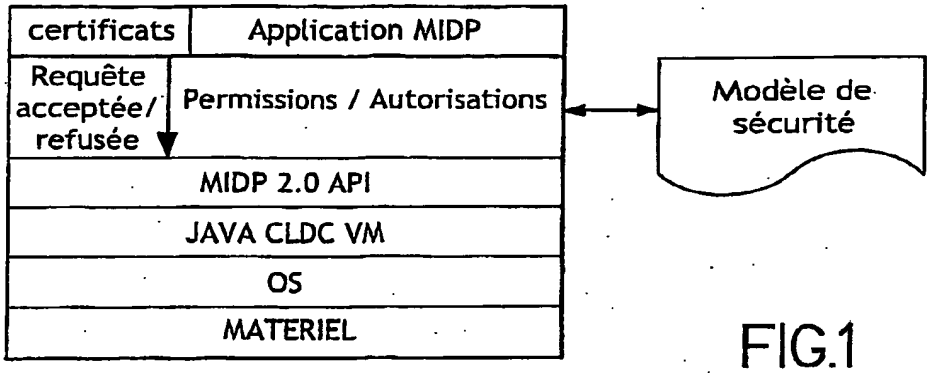

FIG.1

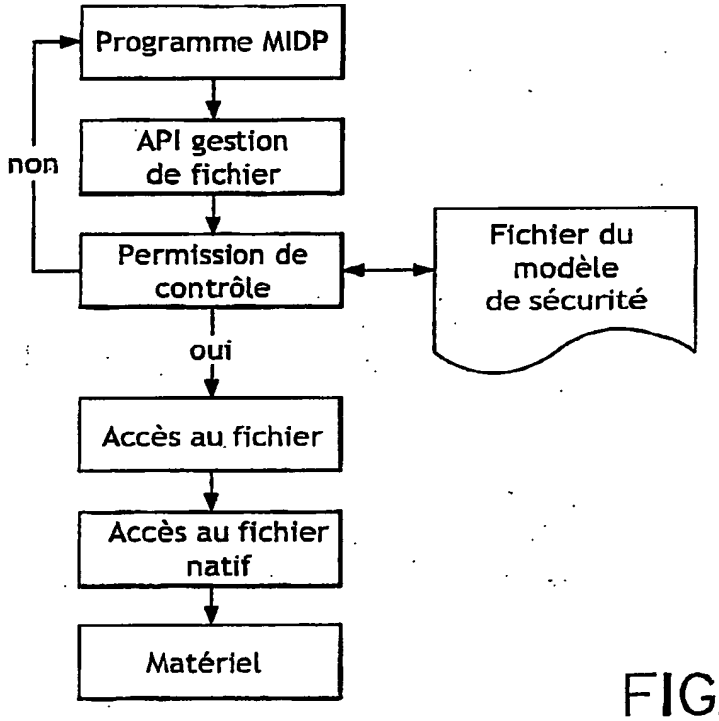

FIG.2

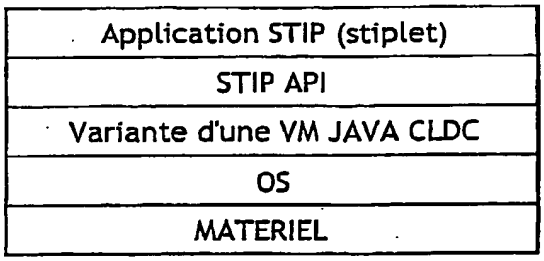

FIG.3

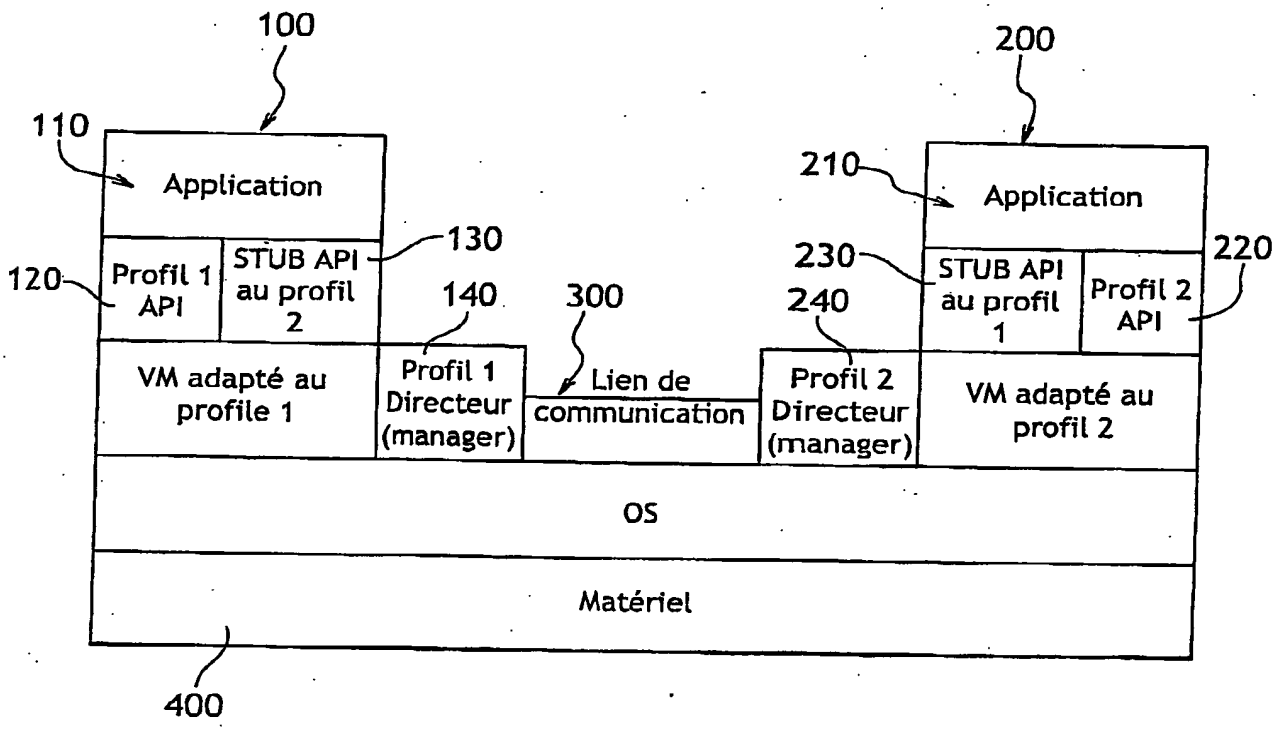
100
200
110
Application
210
Application
130
120
Profil 1 API
STUB API au profil 2
230
STUB API au profil 1
Profil 2 API
220
140
300
240
VM adapté au profile 1
Profil 1 Directeur (manager)
Lien de communication
Profil 2 Directeur (manager)
VM adapté au profil 2
OS
Matériel
400

FIG.4